Design and Implement a Reliable Release Process

Daniel Krzyczkowski
MICROSOFT MVP & SOFTWARE DEVELOPER

@DKrzyczkowski   www.techmindfactory.com

Module Overview

Design and implement release gates and approval processes

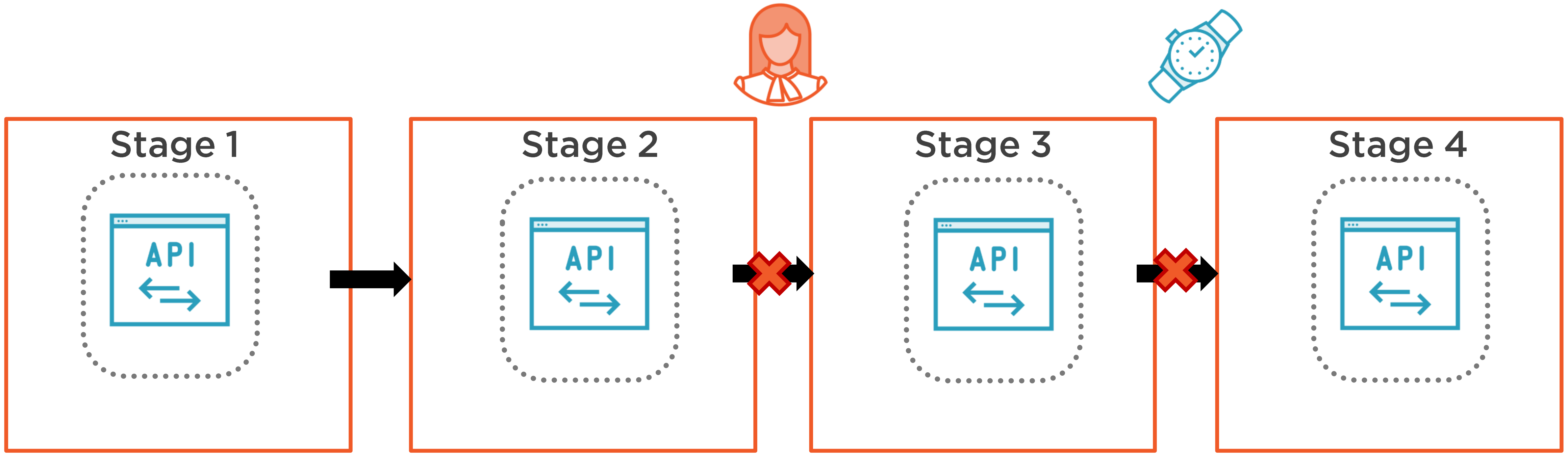
Setup deployment to different cloud environments

Organize shared release configuration and process using multi-stage YAML templates and variable groups

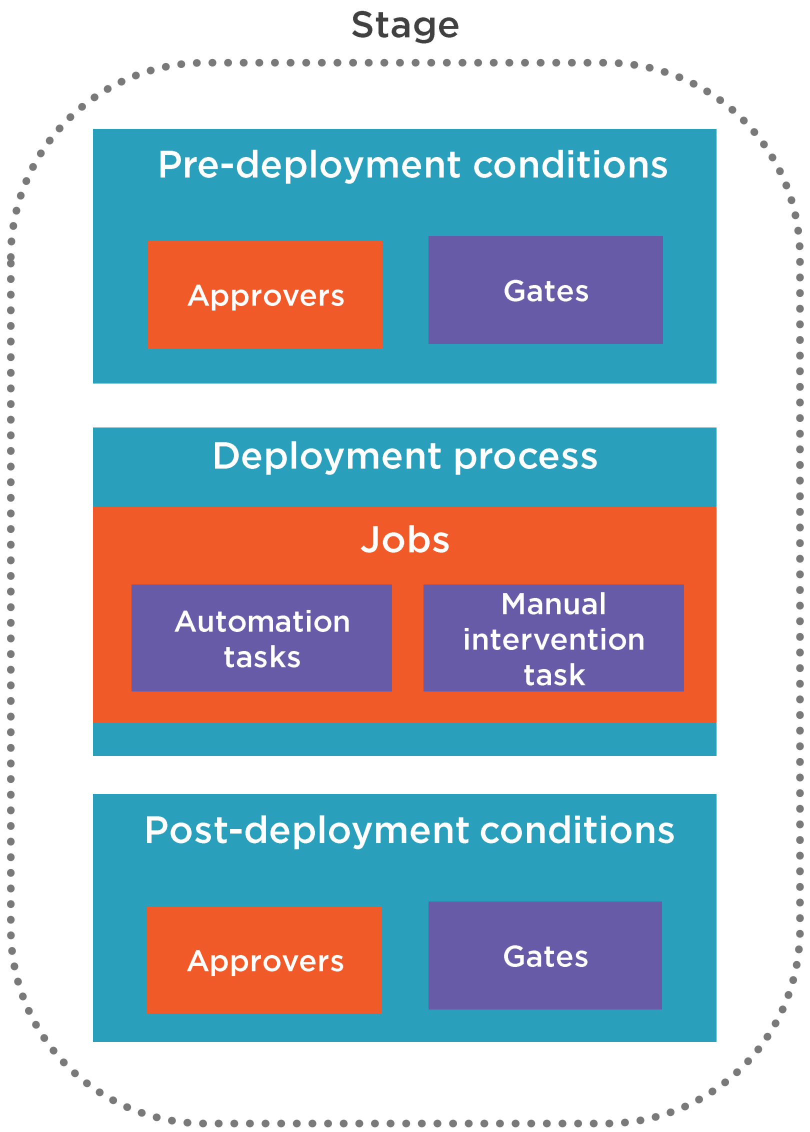
Summary

Organize Shared Release Process with Gates and Approvals

Gates and Approvals



Gates and Approvals



Approvals and gates give you additional control over the start and completion of the release pipeline

Scenarios for Gates and Approvals

- You want to ensure there are no incidents from the monitoring or incident management system for the app after deployment
- Some users must manually validate the change request and approve the deployment to a stage
- During the deployment pipeline, you want to prompt the user to enter a value for a parameter used by the deployment tasks
- Ensure the required status for work items, incidents, and issues is set
- Notify non-Azure Pipelines users such as legal approval departments, auditors, or IT managers

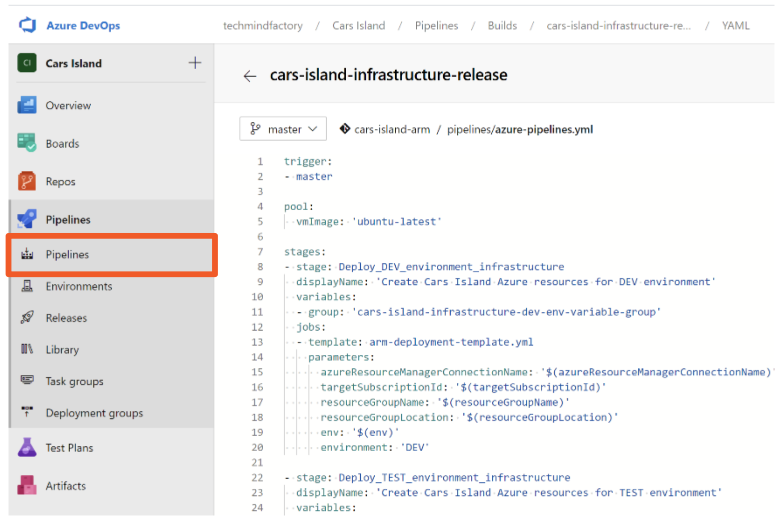
Demo

Implement release gates and approvals

- Update current release with gates and approvals

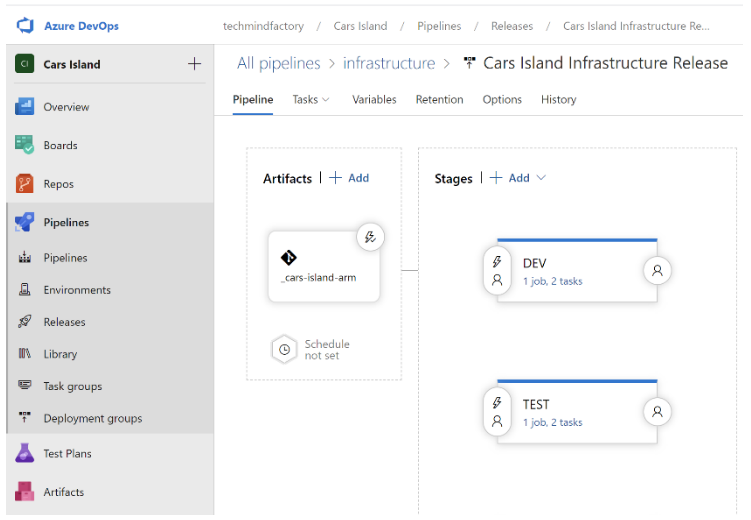
Organize Releases Using Multi-stage YAML Templates

Azure DevOps Pipelines and Releases


Azure DevOps
techmindfactory / Cars Island / Pipelines / Builds / cars-island-infrastructure-re... / YAML
Cars Island
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
Task groups
Deployment groups
Test Plans
Artifacts
cars-island-infrastructure-release
master
cars-island-arm / pipelines/azure-pipelines.yml

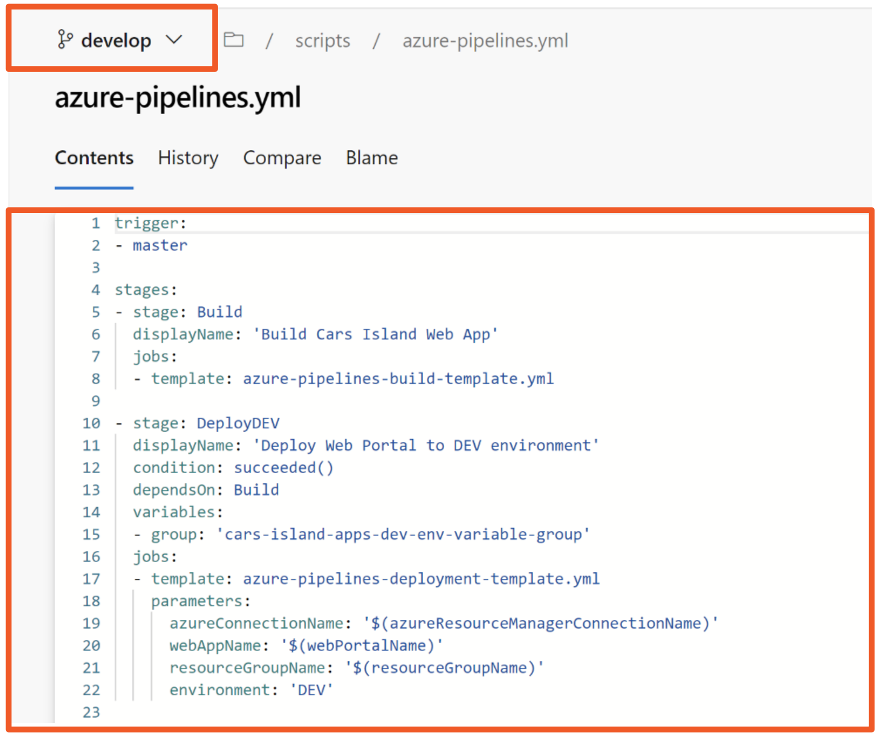
```yaml
trigger:
- master

pool:
  vmImage: 'ubuntu-latest'

stages:
- stage: Deploy_DEV_environment_infrastructure
  displayName: 'Create Cars Island Azure resources for DEV environment'
  variables:
  - group: 'cars-island-infrastructure-dev-env-variable-group'
  jobs:
  - template: arm-deployment-template.yml
    parameters:
      azureResourceManagerConnectionName: '$(azureResourceManagerConnectionName)'
      targetSubscriptionId: '$(targetSubscriptionId)'
      resourceGroupName: '$(resourceGroupName)'
      resourceGroupLocation: '$(resourceGroupLocation)'
      env: '$(env)'
      environment: 'DEV'

- stage: Deploy_TEST_environment_infrastructure
  displayName: 'Create Cars Island Azure resources for TEST environment'
  variables:
```

Pipelines (YAML files)



Releases (Designer)

Define Pipelines Using YAML Syntax



azure-pipelines.yml

```yaml
trigger:
- master

stages:
- stage: Build
  displayName: 'Build Cars Island Web App'
  jobs:
  - template: azure-pipelines-build-template.yml

- stage: DeployDEV
  displayName: 'Deploy Web Portal to DEV environment'
  condition: succeeded()
  dependsOn: Build
  variables:
  - group: 'cars-island-apps-dev-env-variable-group'
  jobs:
  - template: azure-pipelines-deployment-template.yml
    parameters:
      azureConnectionName: '$(azureResourceManagerConnectionName)'
      webAppName: '$(webPortalName)'
      resourceGroupName: '$(resourceGroupName)'
      environment: 'DEV'

```

Build and release pipelines can be defined in a YAML file called azure-pipelines.yml

YAML files can be stored in the source code repository

YAML Files

The pipeline is versioned together with the application's source code

Validation of the changes is done through code reviews in pull requests

Every branch you use can modify the YAML pipeline file and any change might cause a break in the release process

Demo

Implement release configuration and process using YAML files

- Use variable groups to keep shared configuration
- Setup environments
- Implement multi-stage YAML files
- Apply gates and approvals

Summary

How to implement gates and approvals for the release process

How to organize shared release process with multi-stage YAML files

How to design the release pipeline to ensure reliable order of dependency deployments

Thank you!

Daniel Krzyczkowski
MICROSOFT MVP & SOFTWARE DEVELOPER

@DKrzyczkowski www.techmindfactory.com